

Pametni senzor male potrošnje

Turkalj, Matija

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:168:930018>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-03-29**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 494

PAMETNI SENZOR MALE POTROŠNJE

Matija Turkalj

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 494

PAMETNI SENZOR MALE POTROŠNJE

Matija Turkalj

Zagreb, lipanj 2024.

DIPLOMSKI ZADATAK br. 494

Pristupnik: **Matija Turkalj (0119043375)**
Studij: Računarstvo
Profil: Programsko inženjerstvo i informacijski sustavi
Mentor: prof. dr. sc. Hrvoje Mlinarić

Zadatak: **Pametni senzor male potrošnje**

Opis zadatka:

Korištenjem razvojnog sustava tvrtke Nordic Semiconductor zasnovanog na procesoru nRF52840 potrebno je razviti pametan senzor male potrošnje. Senzor mora imati mogućnost mjerenja jedne ili više vrijednosti, te mogućnost slanja pročitane vrijednosti putem Bluetooth Low Energy komunikacije u unaprijed zadanim intervalima. Pročitane vrijednosti potrebno je spremati u memoriju senzora i po potrebi omogućiti čitanje spremljenih vrijednosti korištenjem tehnologije Near Field Communication. Posebnu pozornost obratiti na potrošnju senzora i načine rada u stanju uštede energije.

Rok za predaju rada: 28. lipnja 2024.

Hvala mentoru prof. dr. sc. Hrvoju Mlinariću na ukazanom strpljenju i razumijevanju.

Također bih želio zahvaliti roditeljima, bratu i djevojci Ivani na svojoj ukazanoj potpori kroz cijeli studij. Bez svih vas sve ovo ne bi bilo moguće.

Sadržaj

1. Uvod	2
2. Nordic Semiconductor i sustav na čipu nRF52840	3
2.0.1. Specifikacije nRF52840 procesora	3
2.0.2. Nedostaci uređaja nRF52840	8
3. Tehnologija bežične osobne mreže BLE (eng. "Bluetooth Low Energy") i tehnologija kratkodometne mreže NFC (eng. "Near Field Communication")	9
3.0.1. Osnove tehnologije BLE	9
3.0.2. BLE u nRF52840	10
3.0.3. Osnove NFC tehnologije	11
3.0.4. NFC u nRF52840	14
4. Arhitektura ostvarenog sustava	16
4.0.1. Sklopovlje	16
4.0.2. Programska potpora	19
5. Zaključak	35
Literatura	36
Sažetak	37
Abstract	38

1. Uvod

U današnjem svijetu tehnologije, potreba za sensorima male potrošnje energije postaje sve važnija. S obzirom na rastući trend povezivanja različitih uređaja putem Interneta stvari (IoT), efikasna energetska potrošnja postaje ključna karakteristika. Rad se bavi razvojem pametnog senzora male potrošnje korištenjem razvojnog sustava tvrtke Nordic Semiconductor, zasnovanog na čipu nRF52840. Glavni cilj rada je implementirati senzor koji ne samo da efikasno mjeri i prenosi podatke putem Bluetooth Low Energy (BLE) i Near Field Communication (NFC) komunikacija, već i optimizira svoju energetska potrošnju kroz različite radne modove.

Svrha rada je razvoj pametnog senzora male potrošnje koji može mjeriti vrijednosti, pohraniti te podatke, i prenositi ih putem BLE komunikacije u unaprijed zadanim intervalima. Dodatno, senzor će omogućiti slanje spremljenih vrijednosti korištenjem tehnologija Near Field Communication (NFC) i BLE. Ovaj rad pokazuje praktičnu implementaciju uređaja koji može biti korišten u različitim aplikacijama, od industrijskog nadzora do pametnih kućnih uređaja, te u medicinskoj tehnologiji.

2. Nordic Semiconductor i sustav na čipu nRF52840

Nordic Semiconductor je norveška kompanija specijalizirana za proizvodnju integriranih krugova za bežičnu komunikaciju kratkog dometa. Njihovi proizvodi naširoko se koriste u raznim industrijama, uključujući zdravstvo, pametne kuće, industrijsku automatizaciju i nosive uređaje. Jedan od najistaknutijih proizvoda Nordic Semiconductor-a je nRF52840 procesor, koji je dio njihove nRF52 serije SoC (sustav na čipu eng. "System on Chip") rješenja. Sustav na čipu je integrirani krug koji objedinjuje sve komponente računalnog sustava na jednom čipu. SoC je dizajniran za specifične primjene i može uključivati različite funkcionalne module kao što su procesor, memorija, ulazno-izlazne jedinice, analogno-digitalni pretvarači, komunikacijski moduli i drugi specifični periferni uređaji. Ovaj integrirani pristup omogućuje visoku razinu funkcionalnosti (kao što je povećanje brzine komunikacije između komponentata s obzirom na to da se sve nalaze na istom čipu), manju potrošnju energije, smanjenje veličine uređaja i smanjenje troškova proizvodnje.

2.0.1. Specifikacije nRF52840 procesora

Hardverske specifikacije

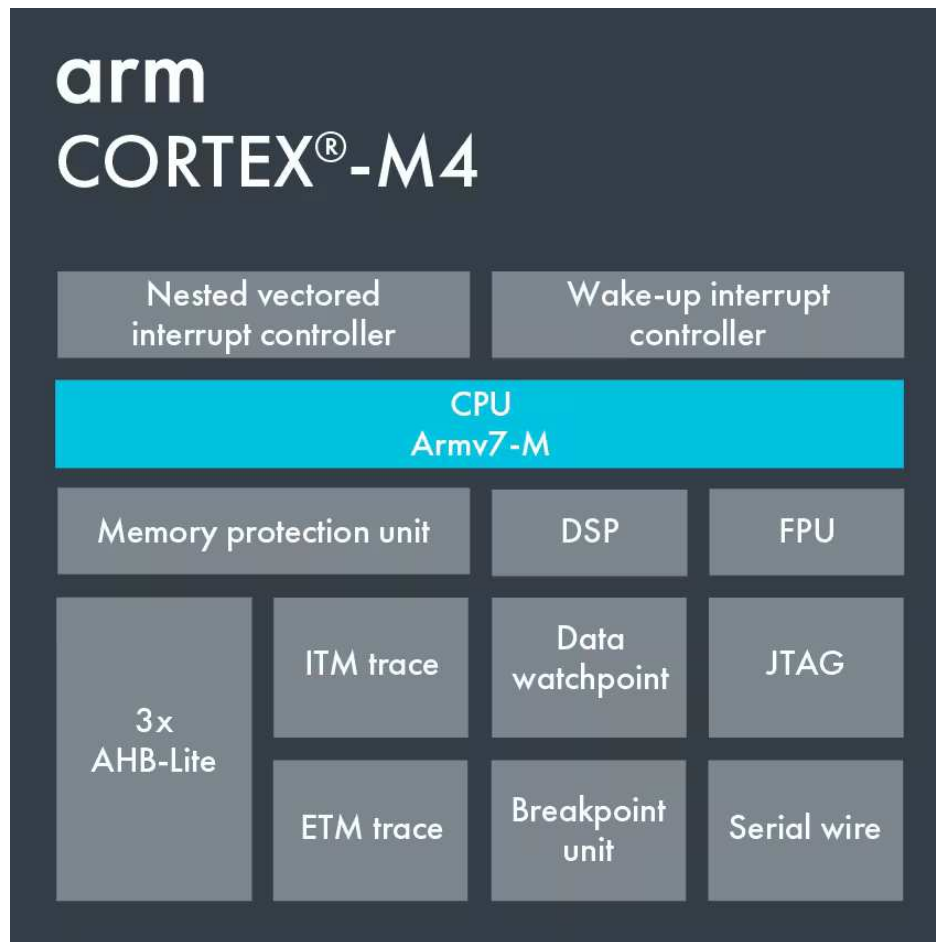
nRF52840 je visoko integrirani SoC koji se temelji na ARM Cortex-M4 procesoru s FPU (eng. "Floating Point Unit"). Radi na frekvenciji do 64 MHz i podržava velik broj perifernih uređaja, što ga čini vrlo svestranim rješenjem za pametne senzore (sustave koji ne samo mjere podatke nego ih potencijalno i obrađuju te šalju raznim komunikacijskim protokolima). Procesor je opremljen s 1 MB flash memorije i 256 KB RAM memorije, što omogućuje dovoljno prostora za složene aplikacije i obradu podataka.

ARM Cortex-M4 procesor, integriran u nRF52840 SoC, predstavlja središnju kompo-

mentu ovog učinkovitog čipa. Cortex-M4 koristi 32-bitnu RISC (eng. "Reduced Instruction Set Computing") arhitekturu, koja omogućuje visoku učinkovitost i performanse uz jednostavnost dizajna i smanjenje troškova proizvodnje. S obzirom da procesor radi na frekvenciji do 64 MHz, on pruža dovoljno računalne snage za obradu složenih zadataka u stvarnom vremenu. Osim toga, integrirana jedinica za rad s pomičnim zarezom (FPU) omogućuje brzu i točnu obradu matematičkih operacija. Procesor je dizajniran s fokusom na energetske učinkovitost, s različitim načinima rada koji omogućuju uštedu energije. To uključuje način rada ultra niske potrošnje (ULP), koji može smanjiti potrošnju energije na minimum kada uređaj nije aktivan, ali i dalje omogućava brzo buđenje i povratak u radno stanje kada je to potrebno.

ARM Cortex-M4 procesor koristi trostupanjski cjevovod (pipeline) za ubrzanje izvršavanja instrukcija, smanjujući kašnjenja i povećavajući protočnost podataka kroz procesor. Odvojene sabirnice za instrukcije i podatke omogućuju simultani pristup memoriji, čime se povećava brzina obrade. Kombinacija 16-bitnih i 32-bitnih instrukcija, poznata kao Thumb-2 instrukcijski set, omogućuje optimizaciju veličine koda i performansi, smanjujući memorijske zahtjeve i povećavajući brzinu izvršavanja.

Napredni sustav za upravljanje prekidima, NVIC (eng. "Nested Vectored Interrupt Controller"), omogućuje brzo i učinkovito rukovanje prekidima, smanjujući latenciju i poboljšavajući responzivnost sustava (slika 2.1 gornji lijevi kut). Na slici 2.1 pored NVIC-a nalazi se WIC (eng. "Wake-up Interrupt Controller"). Ovaj kontroler omogućuje procesoru da se probudi iz stanja niske potrošnje energije kada se dogodi prekid. To je ključno za uređaje koji trebaju biti energetske učinkoviti, jer omogućuje procesoru da bude u stanju mirovanja kada nije aktivno potrebna obrada, čime se štedi energija. U središtu slike nalazi se CPU, središnji procesor koji izvršava instrukcije programa. U ARM Cortex-M4 procesoru, CPU je temeljen na Armv7-M arhitekturi.



Slika 2.1. Arm Cortex-M7 shema[1]

Desno od CPU-a nalazi se MPU (eng. "Memory Protection Unit"). MPU omogućuje zaštitu memorije kontrolirajući pristup različitim dijelovima memorije. To poboljšava sigurnost i stabilnost sustava spriječavanjem neovlaštenog pristupa i potencijalnih grešaka koje bi mogle nastati zbog nepravilnog rukovanja memorijom. Moguće greške koje MPU sprječava ograničavanjem pristupa memoriji su buffer overflow (zapisivanje podataka izlazi iz skupa određenog bufferom), stack overflow, null pointer dereferencing.

Pored MPU-a nalazi se DSP (eng. "Digital Signal Processing") jedinica. DSP instrukcije omogućuju procesoru učinkovitu obradu digitalnih signala. Ovo je korisno za aplikacije koje uključuju audio i video obradu, filtriranje signala, kompresiju podataka i druge zadatke koji zahtijevaju brzu i preciznu obradu podataka.

FPU (eng. "Floating Point Unit") se nalazi desno od DSP-a. FPU omogućuje procesoru obradu operacija s pomičnim zarezom (eng. "floating-point calculations") mnogo brže i točnije nego što bi to bilo moguće sa softverskom emulacijom. Ovo je posebno ko-

risno za znanstvene izračune, digitalnu obradu signala i druge aplikacije koje zahtijevaju visok stupanj preciznosti.

Na donjoj strani slike prikazani su moduli za praćenje i otklanjanje pogrešaka. "ITM Trace" (Instrumentation Trace Macrocell) omogućuje praćenje izvršavanja programa i prikupljanje podataka o performansama. Ovo je korisno za razvojne inženjere koji trebaju detaljno analizirati rad programa i optimizirati performanse. "ETM Trace" (Embedded Trace Macrocell) pruža napredne mogućnosti praćenja izvršavanja instrukcija u stvarnom vremenu, što omogućuje vrlo detaljno praćenje i analizu rada programa.

"Data Watchpoint and Trace" (DWT) omogućuje postavljanje točaka nadzora (watchpoints) i praćenje specifičnih podataka tijekom izvršavanja programa. Ovo pomaže u detekciji i otklanjanju grešaka te optimizaciji programa. "JTAG" (Joint Test Action Group) je standard za testiranje i otklanjanje grešaka u integriranim krugovima. Omogućuje pristup internim registrima procesora i memoriji za potrebe testiranja, otklanjanja grešaka i programiranja.

"Breakpoint Unit" omogućuje postavljanje točaka prekida (breakpoints) u programu kako bi se zaustavilo izvršavanje na određenim linijama koda. Služi za otklanjanje grešaka tim što omogućuje detaljno ispitivanje stanja sustava u ključnim točkama programa. "Serial Wire" je sučelje za komunikaciju koje omogućuje brzi prijenos podataka između procesora i vanjskih uređaja ili alata za razvoj, koristeći se minimalnim utjecajem na performanse.

Konačno, "3x AHB-Lite" (Advanced High-performance Bus Lite) omogućuje visoko učinkovitu komunikaciju između procesora i drugih dijelova sustava, kao što su memorija i periferni uređaji. U Cortex-M4 procesoru nalaze se tri AHB-Lite sabirnice koje omogućuju brzi prijenos podataka unutar sustava.

Navedene značajke čine ARM Cortex-M4 procesor u nRF52840 SoC-u izvanrednim izborom za razne ugrađene sustave i IoT aplikacije. Korištenjem ovog procesora, dizajneri i inženjeri mogu stvoriti visoko funkcionalne i pouzdane uređaje koji zadovoljavaju zahtjeve modernih tržišta. Opširniji opis dostupan je na [[2]]).

Funkcionalnosti i prednosti

nRF52840 sustav na čipu nudi brojne funkcionalnosti koje ga čine idealnim za primjene u pametnim sensorima. Prije svega, podrška za Bluetooth 5 omogućuje dug domet (na otvorenom prostoru i do 200 metara, a u zatvorenim prostorima do maksimalno 50 metara), veliku brzinu prijenosa podataka i poboljšanu učinkovitost energije. Bluetooth 5 također omogućuje korištenje tehnologije Bluetooth Low Energy (BLE), koja je ključna za uređaje male potrošnje energije. Najveća prednost Bluetootha 5 zbog koje je BLE efikasniji su veće brzine prijenosa podataka koja je dvostruko veća od ranije verzije jer samim povećanjem brzine prijenosa se troši manje energije na pasivan rad uređaja za odašiljanje Bluetooth signala.

Jedna od ključnih značajki nRF52840 je podrška za značajan broj protokola bežične komunikacije, uključujući Bluetooth 5, Bluetooth Mesh, Thread, Zigbee, 802.15.4, ANT i proprietarne 2.4 GHz protokole. Raznovrsnost komunikacijskih protokola omogućuje širok spektar primjena i interoperabilnost s raznim uređajima i standardima.

Jedan od važnih aspekata primjene nRF52840 u pametnim sensorima je sposobnost podržavanja većeg broja bežičnih protokola, što omogućuje fleksibilnost u dizajnu i interoperabilnost s drugim uređajima i mrežama. Na primjer, senzori koji koriste BLE mogu se lako integrirati s pametnim telefonima i drugim BLE uređajima, dok podrška za Zigbee i Thread omogućuje povezivanje u mrežama za pametne domove i industrijske primjene.

Dodatne funkcionalnosti uključuju podršku za USB 2.0, razne ADC (eng. "Analog-to-Digital Converter") kanale za obradu analognih signala, i bogatu ponudu perifernih uređaja kao što su SPI (eng. "Serial Peripheral Interface"), I2C (eng. "Inter-Integrated Circuit"), UART (eng. "Universal Asynchronous Receiver/Transmitter"), PWM (eng. "Pulse Width Modulation"), i GPIO (eng. "General-Purpose Input/Output"). Ova svestranost omogućuje lako povezivanje i integraciju raznih senzora i drugih komponenti.

Primjene u pametnim sensorima

Sustav na čipu nRF52840 može se široko koristiti u raznim aplikacijama koje zahtijevaju bežičnu komunikaciju i nisku potrošnju energije. U pametnim sensorima procesor omo-

gućuje pouzdano prikupljanje podataka i njihovo bežično slanje u unaprijed definiranim intervalima. Primjeri uključuju zdravstvene uređaje za praćenje vitalnih znakova, industrijske senzore za praćenje okolišnih uvjeta, i senzore za pametne domove koji prate temperaturu, vlagu, prisutnost i druge parametre. Samo neki od primjera upotrebe procesora nRF52840 mogu se pronaći i u primjerima koda koji je dostupan u službenoj dokumentaciji [3]. Neki od primjera su beacons za prepoznavanje blizine i pokreta, monitori otkucaja srca, sustavi za bežični pristup i sustavi plaćanja (oboje koriste NFC) te mnoge druge IoT ili druge primjene.

2.0.2. Nedostaci uređaja nRF52840

Sustav na čipu nRF52840 sadrži i neke nedostatke, na primjer troši više električne energije u usporedbi s nekim konkurentnim mikrokontrolerima. Prema povratnim informacijama korisnika na platformama kao što je Nordic Developer Zone, neki programeri su primijetili da nRF52840 ima veću potrošnju energije u aktivnom stanju i stanju mirovanja u usporedbi sa svojim prethodnikom, nRF52832. Također kako bi se moglo razvijati ovaj uređaj potrebno je upoznati se sa ne tako jednostavim SDK-om (eng. "Software Development Kit") koji za razliku od nekih konkurenata (kao što je npr. uređaj CC2640R2F od tvrtke Texas Instruments koji je sličan uređaju nRF52840, a za koji korisnici tvrde da je lakše razumljiv).

3. Tehnologija bežične osobne mreže BLE (eng. "Bluetooth Low Energy") i tehnologija kratkodometne mreže NFC (eng. "Near Field Communication")

Tehnologija bežične osobne mreže BLE (eng. "Bluetooth Low Energy"), također poznat kao Bluetooth Smart, je tehnologija bežične osobne mreže dizajnirana i optimizirana za aplikacije koje zahtijevaju prijenos podataka na niskoj brzini s minimalnom potrošnjom energije. BLE je razvijen kao dio Bluetooth 4.0 specifikacije i namijenjen je za aplikacije u zdravstvu, fitnessu, beaconima, sigurnosnim uređajima i kućnoj automatizaciji kao i u svim ostalim sustavima u kojima je energetska učinkovitost kritična.

3.0.1. Osnove tehnologije BLE

Jedna od glavnih razlika između BLE-a i klasičnog Bluetootha (znanog i kao Bluetooth Classic) leži u načinu na koji komuniciraju i koliko energije troše. Klasični Bluetooth je dizajniran za kontinuirani prijenos podataka s velikom brzinom, što ga čini pogodnim za aplikacije poput prijenosa zvuka (slušalice), prijenosa datoteka i sličnih zadataka koji zahtijevaju veće brzine prijenosa podataka. Nasuprot tome, BLE je optimiziran za periodični prijenos malih količina podataka uz minimalnu potrošnju energije, čime omogućuje dugotrajnu upotrebu baterija, što često uzrokuje korištenje uređaja i do nekoliko godina na jednoj bateriji. Bluetooth Low Energy ističe nekoliko ključnih karakteristika koje ga čine idealnim za senzore i druge uređaje male potrošnje. Prva i najvažnija prednost BLE-a je niska potrošnja energije. BLE uređaji mogu biti u stanju mirovanja većinu vremena i aktivirati se samo kada je potrebno prenijeti podatke, čime se značajno štedi energija. Druga ključna karakteristika je brzo uspostavljanje veze. BLE koristi vrlo

kratke veze, u trajanju od nekoliko milisekundi, za prijenos podataka, što dodatno smanjuje vrijeme rada i potrošnju energije. Osim toga, BLE podržava visoku gustoću mreže, omogućujući veliki broj uređaja u istoj mreži. Ovo svojstvo je posebno korisno u primjenama poput pametnih zgrada, gdje je potrebna skalabilnost za povezivanje mnogih uređaja. BLE također nudi veliku fleksibilnost u smislu intervala oglašavanja i povezivanja. Ovi intervali mogu se prilagoditi specifičnim potrebama aplikacije, omogućujući optimalnu ravnotežu između performansi i potrošnje energije. BLE proces uparivanja započinje dogovorom između dvaju uređaja o načinu razmjene ključeva kako bi se uspostavila šifrirana veza. Metode uključuju Just Works, Out of Band (OOB) i Passkey Entry, koje nude različite razine sigurnosti protiv napada posrednika (eng. "Man in the middle"). Just Works pruža osnovnu sigurnost bez autentifikacije, dok OOB koristi vanjski kanal za poboljšanu sigurnost, a Passkey Entry uključuje unos zajedničke tajne za snažnu zaštitu. Nakon razmjene ključeva i autentifikacije, uređaji uspostavljaju šifriranu vezu koristeći kratkoročni ključ (eng. "Short-term key", STK). Zatim generiraju dugoročni ključ (eng. "Long-term key", LTK) za buduće sigurne komunikacije. Tijekom faze povezivanja ključevi se pohranjuju u memoriju uređaja, omogućujući sigurno ponovno povezivanje bez ponavljanja procesa uparivanja. Za osiguranje sigurnog prijenosa podataka, BLE koristi AES-CCM enkripciju, osiguravajući povjerljivost, integritet i autentičnost. Osim toga, BLE uređaji periodično mijenjaju svoje adrese uređaja kako bi spriječili praćenje identiteta i prisluškivanje[4].

3.0.2. BLE u nRF52840

nRF52840 je dizajniran da potpuno podrži BLE tehnologiju, uključujući najnovije verzije Bluetooth 5 specifikacije. To uključuje podršku za dug doomet, veću brzinu prijenosa podataka i veće oglašavanje kapaciteta. Implementacija BLE-a na nRF52840 koristi bogati skup programskih knjižnica i alata koje pruža Nordic Semiconductor, uključujući SoftDevice, što je programski stog (eng. "stack") koji omogućuje funkcionalnost BLE.

Jedna od ključnih prednosti BLE-a je mogućnost prilagođavanja intervala slanja podataka kako bi se optimizirala potrošnja energije. Na nRF52840, intervali oglašavanja i povezivanja mogu se dinamički mijenjati u ovisnosti o potrebama aplikacije. Na primjer, uređaj može rijetko oglašavati dok je u stanju mirovanja, a zatim povećati učestalost oglašavanja kada detektira aktivnost ili kada je potrebno prenijeti kritične podatke.

BLE uređaji uz pomoć senzora, koji mogu kontinuirano mjeriti neke parametre kao što su temperatura, vlaga, tlaka ili pokret, donose funkcionalnost povremenog slanja tih podatke nekom centralnom uređaju (npr. pametnom telefonu, računalu ili gatewayu). Ovi podaci mogu se koristiti za različite analize, obrade, kontrolu ili prikazivanje korisnicima. Na primjer, pametni termostat može koristiti BLE za prikupljanje podataka o temperaturi iz različitih soba i na temelju tih podataka prilagođavati rad grijanja ili hlađenja i u isto vrijeme obaviještavati korisnika o trenutnom stanju temperature.

Optimizacija potrošnje energije je ključna za dugotrajan rad senzora na bateriju. BLE omogućava nekoliko strategija za smanjenje potrošnje energije:

- Korištenje dužih intervala spavanja: Uređaji mogu duže vremena provoditi u stanju mirovanja i kratko se buditi samo za prijenos podataka.
- Smanjenje snage oglašavanja: Smanjenje snage signala oglašavanja može značajno smanjiti potrošnju energije, posebno u aplikacijama gdje je domet manje bitan, što je u svijetu senzora i IoT-a nerijedak slučaj.

Kao dobar primjer gdje se rijetko, ali kontinuirano odvija slanje podataka ističe se očitavanje potrošnje brojila u kućanstvu ili poslovnom objektu. Takvo očitavanje vrši se svega jednom mjesečno i ako se

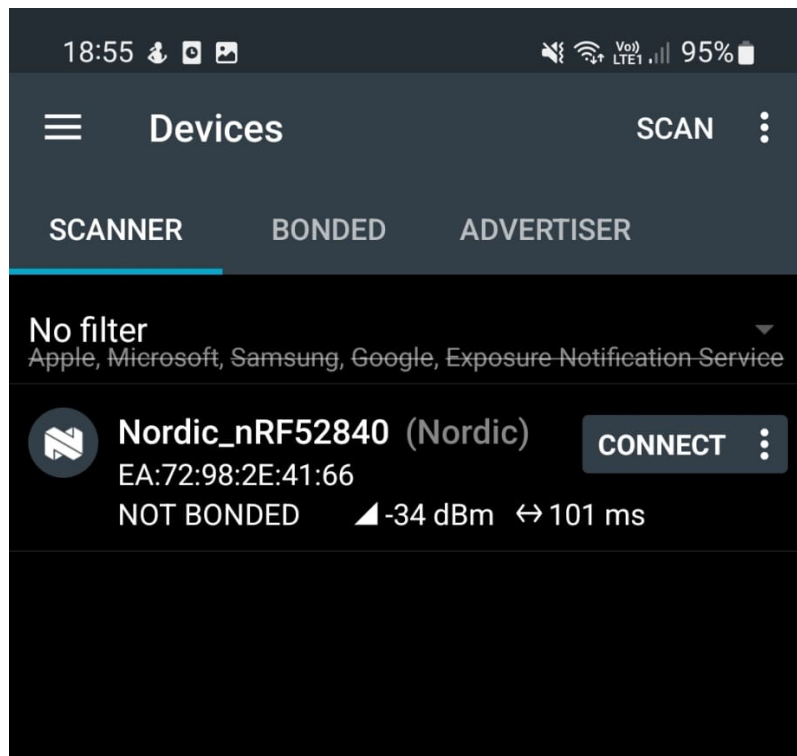
3.0.3. Osnove NFC tehnologije

Kratkodometna tehnologija NFC (eng. "Near Field Communication") je tehnologija koja omogućuje dvama elektroničkim uređajima, obično mobilnom uređaju i drugom uređaju, uspostavljanje komunikacije približavanjem unutar nekoliko centimetara. NFC radi na frekvenciji od 13,56 MHz i podržava brzine prijena podataka od 106 kbit/s do 424 kbit/s. NFC tehnologija se temelji na induktivnom spajanju dviju antena koje rade na frekvenciji koja pripada globalno dostupnom i nelicenciranim ISM (industrijskom, znanstvenom i medicinskom) radio pojasu što osigurava da NFC uređaji mogu raditi diljem svijeta bez problema s interferencijom. NFC uređaji mogu raditi u tri različita načina. U načinu rada od uređaja do uređaja (eng. "peer to peer"), dva NFC uređaja izravno komuniciraju međusobno kako bi razmijenili informacije, što se obično koristi za funkcionalnosti poput dijeljenja datoteka ili postavljanja Bluetooth i Wi-Fi veza između

uređaja. U načinu rada čitaj/piši (eng. "read/write"), NFC uređaj, obično pametni telefon, čita informacije s NFC oznake ili upisuje podatke na nju, što se koristi u aplikacijama kao što su pametni plakati ili elektronske vizitke, gdje korisnik dodiranjem telefona na NFC oznaku dohvaća informacije. U načinu emulacije kartice, NFC uređaj emulira pametnu karticu, omogućujući mu korištenje postojeće beskontaktna infrastrukture za primjene poput plaćanja ili kontrole pristupa.

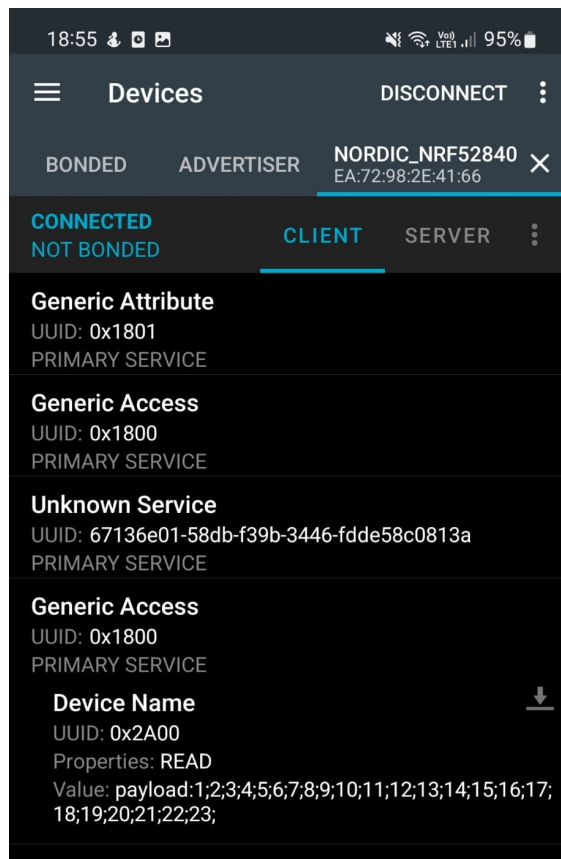
NFC komunikacija može uključivati različite formate razmjene podataka, a najčešći je NDEF (eng. "NFC Data Exchange Format"), standardizirani format podataka koji omogućuje pohranu i razmjenu informacija između NFC-kompatibilnih uređaja. NDEF poruka sastoji se od jednog ili više NDEF zapisa. Svaki zapis sadrži zaglavlje i sadržaj. Zaglavlje specificira vrstu, duljinu i strukturu sadržaja, dok sadržaj nosi stvarne podatke, poput URL-ova, običnog teksta ili medijskih datoteka tipa MIME. NDEF podržava različite vrste zapisa, uključujući dobro poznate tipove (kao što su tekst i URI), medijske tipove (kao što su slike i videozapisi) i vanjske tipove (za prilagođene aplikacije). NDEF poruke mogu se pohraniti na NFC tagove, ugraditi u pametne postere ili izravno razmjenjivati između NFC uređaja kroz peer-to-peer komunikaciju. Kada NFC čitač komunicira s NFC tagom ili drugim NFC uređajem, čita NDEF poruku, interpretira zapise i obrađuje sadržaje u skladu s tim. Sigurnost je ključni aspekt NFC tehnologije, posebno u aplikacijama poput beskontaktnih plaćanja. NFC po dizajnu kratkog dometa ima sigurnosnu prednost, ali to nije uvijek dovoljno. Osjetljive informacije razmijenjene tijekom transakcija mogu se šifrirati kako bi se spriječilo presretanje, a za aplikacije visokog stupnja sigurnosti poput mobilnih plaćanja, NFC uređaji mogu uključivati sigurni element (eng. "Secure Element" - SE), namjensku hardversku komponentu dizajniranu za sigurno pohranjivanje kriptografskih ključeva i izvršavanje sigurnih transakcija. Ova komponenta danas je široko korištena u pametnim telefonima.

Slika 3.1. prikazuje nRF Connect aplikaciju kojom je moguće pristupiti uređaju nRF52840 putem BLE mreže.



Slika 3.1. Korišteni uređaj na BLE mreži

Slika 3.2. prikazuje komunikaciju mobilnog uređaja i ostvarenog sustava korištenjem BLE tehnologije. U gornjem desnom kutu vidljiv je naziv uređaja, a na dnu slike vidljiv je i poslani set podataka.



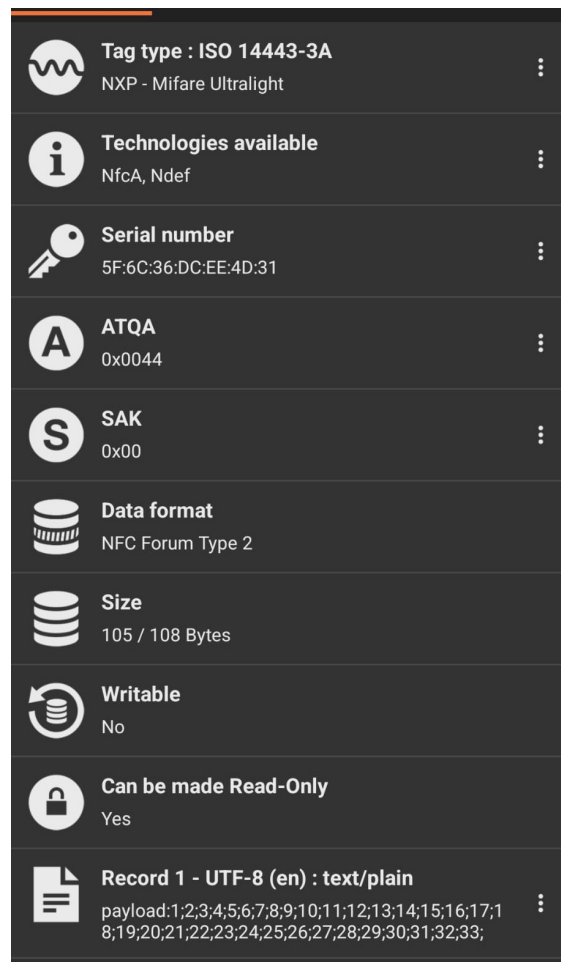
Slika 3.2. Dohvat podataka preko BLE

3.0.4. NFC u nRF52840

Sustav na čipu nRF52840 integrira mogućnosti korištenja tehnologije NFC. Uređaj podržava NFC-A tag funkcionalnost, što mu omogućuje da djeluje kao NFC oznaka koja može komunicirati s NFC čitačima, uključujući pametne telefone i specijalizirane NFC uređaje. Ova funkcionalnost implementirana je pomoću NFC kontrolera na čipu, koji upravlja modulacijom i demodulacijom signala, osiguravajući pouzdanu komunikaciju. NFC na nRF52840 može se koristiti za različite svrhe, kao što su pojednostavljenje uparivanja Bluetooth uređaja, omogućavanje sigurnih transakcija i slanje skupa podataka. NFC modul na nRF52840 dizajniran je da bude energetske učinkovit, što ga čini pogodnim za uređaje koji rade na baterije. Čip može detektirati NFC polje i probuditi se iz niskog energetske stanja kako bi izvršio potrebne operacije, čime se štedi energija. nRF52840 također uključuje programabilni periferni međusklop (PPI) koji omogućuje NFC događajima da aktiviraju druge periferne uređaje bez intervencije CPU-a, što dodatno povećava energetske učinkovitost. Na primjer, NFC događaj detekcije može izravno aktivirati GPIO pin za buđenje sustava ili iniciranje Bluetooth uparivanja. Nordic

Semiconductor nudi sveobuhvatan SDK za nRF52840, koji uključuje knjižnice i primjere za implementaciju NFC funkcionalnosti. Ovaj SDK pruža alate za konfiguriranje NFC oznaka, rukovanje NFC događajima i integraciju NFC-a s drugim funkcijama sustava poput BLE-a.

Na slici 3.3. prikazana je poruka koja je poslana sa ostvarenog sustava korištenjem NFC tehnologije. Na slici se vide metapodaci o ostvarenoj komunikaciji te sami sadržaj poruke na samom dnu slike. Aplikacija koja je korištena za očitavanje senzora zove se NFC Tools i moguće ju je preuzeti na sljedećoj lokaciji: [5]



Slika 3.3. NFC set podataka poslan iz ostvarenog sustava prikazan u aplikaciji NFC Tools

4. Arhitektura ostvarenog sustava

Ostvareni sustav sastoji se od sklopovlja i programske potpore (eng. "software"). Korišteno sklopovlje sastoji se od sustava na čipu nRF52840 te NFC antene, a programska potpora je pisana u jeziku C uz pomoć korištenja razvojnog alata Visual Studio Code te ekstenzija koje je napravila tvrka Nordic Semiconductor.

4.0.1. Sklopovlje

Korištena NFC antena koja simulira oznaku nalazi se na slici 4.1. Kapacitet simulirane oznake iznosi 1024 bytea te stoga predstavlja usko grlo u prijenosu podataka putem NFC tehnologije. NFC antena je postavljena u blizini nRF52840 ploče kako bi omogućila lako dohvaćanje podataka pomoću NFC čitača, kao što je pametni telefon. Antena ne zahtijeva dodatno napajanje jer koristi energiju emitiranu od strane NFC čitača za rad.

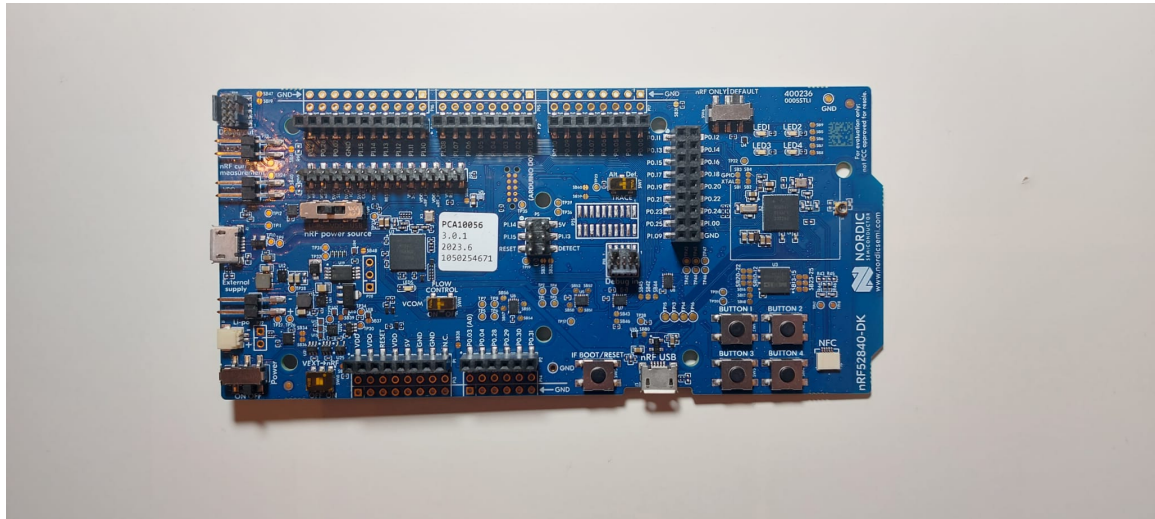


Slika 4.1. NFC antena

Na slici 4.2 nalazi se sustav na čipu nRF52840 koji se sastoji od mnogo komponenti. Glavne komponente su Micro-USB port koji se koristi za napajanje, programiranje i komunikaciju. USB-UART most nudi USB na UART sučelje. Za programiranje i debugiranje, ploča uključuje ugrađeni Segger J-Link OB debugger i Debug Out konektor. Memorija je dostupna putem QSPI Flasha, što pruža dodatni prostor za pohranu. Ploča također sadrži četiri korisnički programabilna LED svjetla, četiri korisnički programabilna gumba i jedan reset gumb za korisničku interakciju. Oscilatori uključuju 32 MHz kristalni oscilator za glavni sistemski sat i 32.768 kHz kristalni oscilator za sat u pravom vremenu (eng. "real-time clock"). Integrirana PCB antena osigurava bežičnu komunikaciju, s opcijom povezivanja vanjske antene. Iako uređaj nema integrirane senzore moguće ih je priključiti na bilo koji od 48 programabilnih GPIO pinova. Za spajanje NFC antene uređaj nudi poseban port.

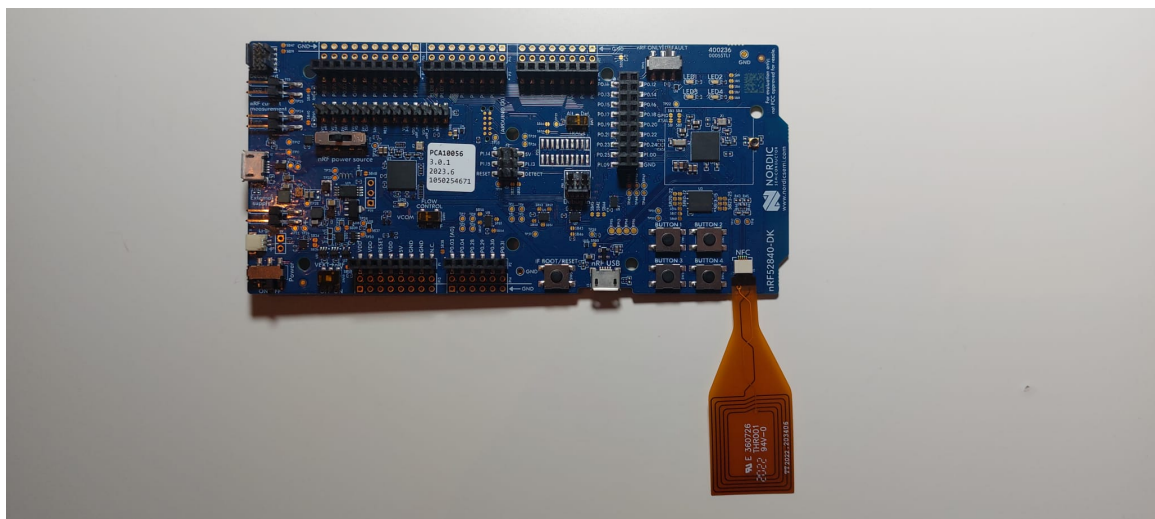
Od spomenutih komponenata korištene su flash memorija, napajanje, LED svjetla,

port za NFC antenu te sistemski sat. U sredini s lijeve strane nalazi se ulaz za napajanje te prijenos programskog koda koji se izvršava na uređaju, ulaz je tipa microusb. Na gornjoj desnoj strani uređaja nalaze se 4 LED lampice te na krajnoj desnoj donjoj strani nalazi se ulaz za NFC antenu u koji je bila uključena antena sa slike 4.1, a prikaz ostvarenog sustava nalazi se na slici 4.3.



Slika 4.2. nRF52840

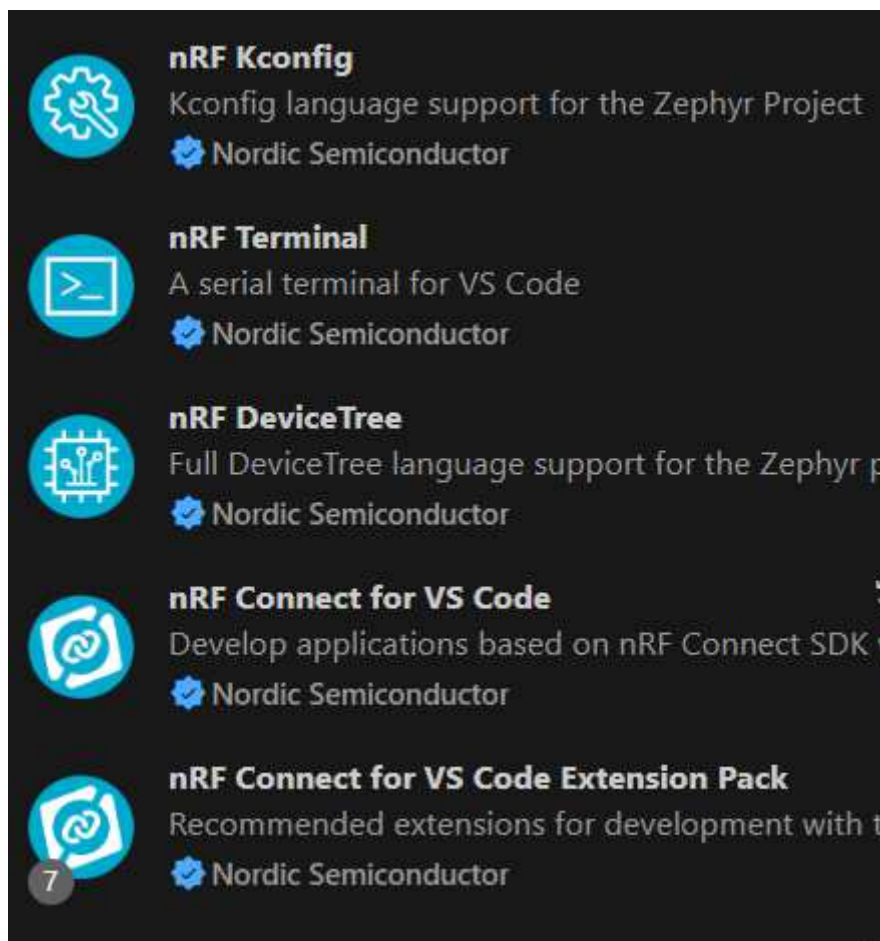
Sustav na čipu nRF52840 koji se nalazi na slici 4.3 pokrenut je napajanjem iz male CR2032 litijske baterije, a također ga je moguće spojiti na napajanje korištenjem micro usb kabela koji je spojen na neki izvor električne struje (kao što je npr. laptop).



Slika 4.3. Sustav na čipu nRF52840 sa spojenom NFC antenom

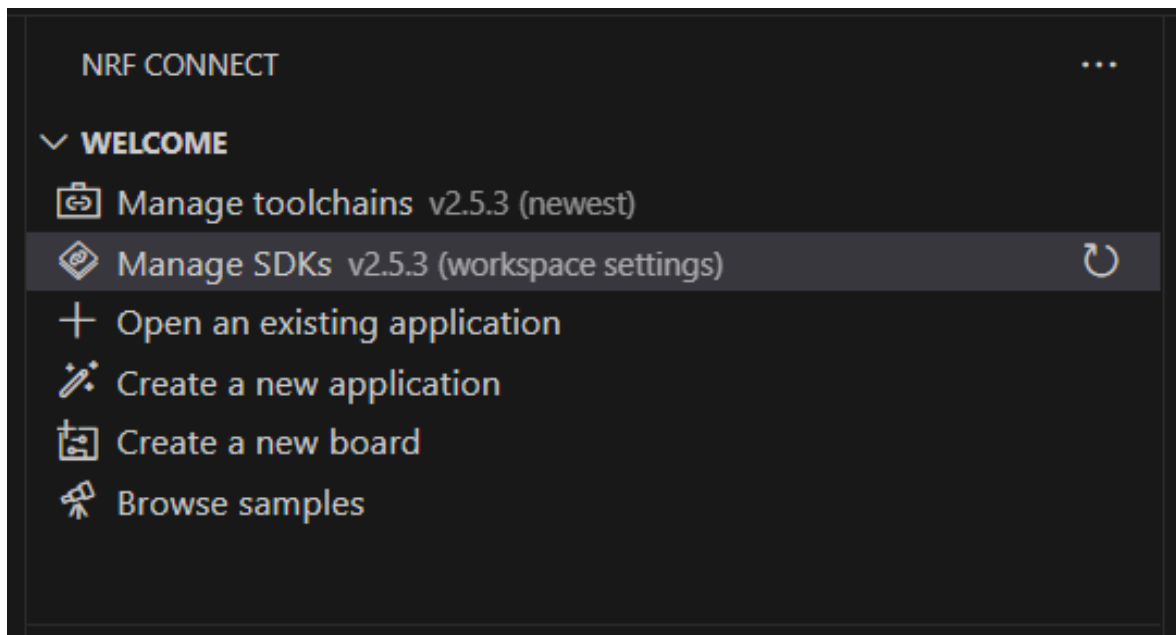
4.0.2. Programska potpora

Razvojno okruženje Za razvijanje programskog koda korišten je besplatni, programsko okruženje otvorenog koda (eng. "open source") Visual Studio Code (VSC). Odabrani uređivač korišten je primarno iz razloga što tvrtka Nordic Semiconductor izrađuje i održava nastavke (eng. "extensions") za ovaj uređivač koji znatno olakšavaju programiranje na uređaju nRF52840. Skup svih alata koji su dostupni nalaze se na slici 4.5. Instaliranjem glavnog alata nRF Connect for VS Code Extension Pack (na slici 4.5 skroz na dnu), instaliraju se i svi ostali alati. Od dostupnih alata najbitnije je istaknuti nRF Terminal koji prikazuje sve izlazne informacije za vrijeme build i flash faza razvijanja projekta. nRF Kconfig pruža intuitivniji GUI (grafičko sučelje) za KConfig strukturu, međutim nije korišten za razvoj ovog projekta kao ni alat nRF DeviceTree jer su inicijalne postavke bile i više nego dovoljne. Na slici 4.4 spominje se Zephyr. Zephyr je operativni sustav u stvarnom vremenu RTOS (engl. "Real-time operating system") otvorenog koda. Dizajniran je za ugradbene računalne sustave. Razvijen pod Linux Foundation, Zephyr je optimiziran za uređaje s ograničenim resursima. Ključne tehničke značajke uključuju preemptivni, priorizirani raspored, koji omogućuje precizno upravljanje zadacima i performanse u stvarnom vremenu. Zephyr podržava aplikacije s jednom ili više dretvi (eng. "thread"), koristeći lagane mehanizme za komunikaciju između niti poput semafora, muteksa i redova poruka. Zephyr je visoko modularan i konfigurabilan, što omogućuje razvojnim programerima da uključe samo potrebne komponente, optimizirajući memorijske i procesorske resurse. Za razvoj, Zephyr se integrira s alatnim lancima poput GCC-a i pruža opsežnu podršku za debugiranje i testiranje, uključujući jedinice za testiranje i simulacijska okruženja. Projekt koristi snažnu zajednicu i opsežnu dokumentaciju, potičući kolaborativno razvojno okruženje.



Slika 4.4. VSC i alati za rad Nordic Semiconductor proizvodima

Alat nRF Connect for VS Code najviše je korišten prilikom izrade ovog rada. Kako bi se mogao koristiti ovaj alat potrebno je u alatu otvoriti projektni kod koji se namjerava prenijeti na uređaj nRF52840 i namjestiti odgovarajući toolchain i SDK (slika 4.6). Obično je u redu koristiti najnoviji mogući i ne bi trebalo doći ni do kakvih problema pri pokretanju sustava. Jedino je bitno da verzije budu usklađene. U ovom radu korištene su verzije v2.5.3 toolchaina i SDK-a.

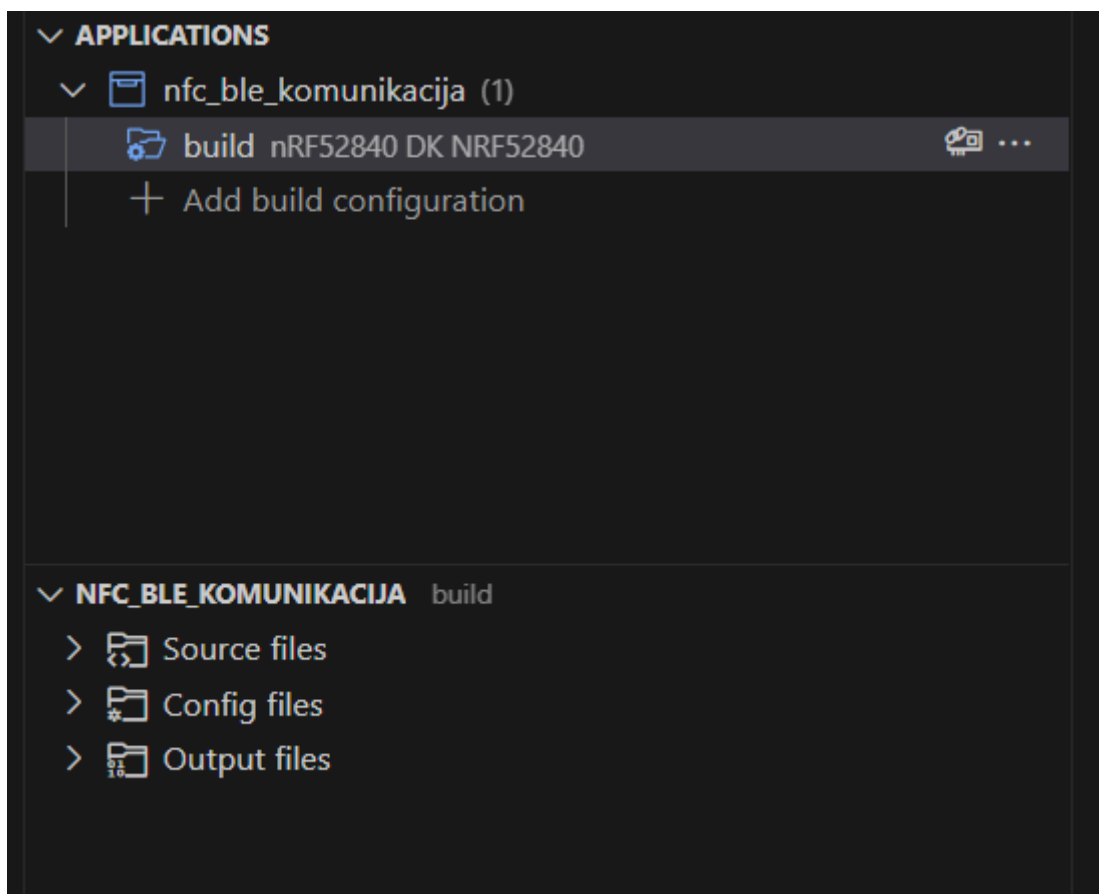


Slika 4.5. nRF Connect

Nakon što su namješteni toolchain i SDK potrebno je odrediti postavke prema kojima će se odraditi build faza projekta (slika 4.7). U build postavkama potrebno je dodati novu konfiguraciju. Najbitnije postavke koje treba postaviti su razvojne pločice za koje je će biti izgrađen (eng. "build") kod, što je u ovom slučaju nRF52840 te ukoliko je potrebno definiranje konfiguracijskih datoteka. U ovom slučaju bilo je potrebno definirati konfiguracijsku datoteku zbog korištenja tehnologija BLE, NDEF i NFC, ali i zbog lakšeg i preglednijeg grupiranja nekih postavaka. Najznačajnije postavke prikazane su na slici 4.7. Na slici su slovima "y" označeni pozitivno pridružene vrijednosti, nalik na primitivni tip boolean. U datoteci je definirano kako je dopušten reboot uređaja, korištenje flash memorije, korištenje NDEF formata i BT komunikacije te je postavljeno ime uređaja koje će se prikazivati na BLE mreži.

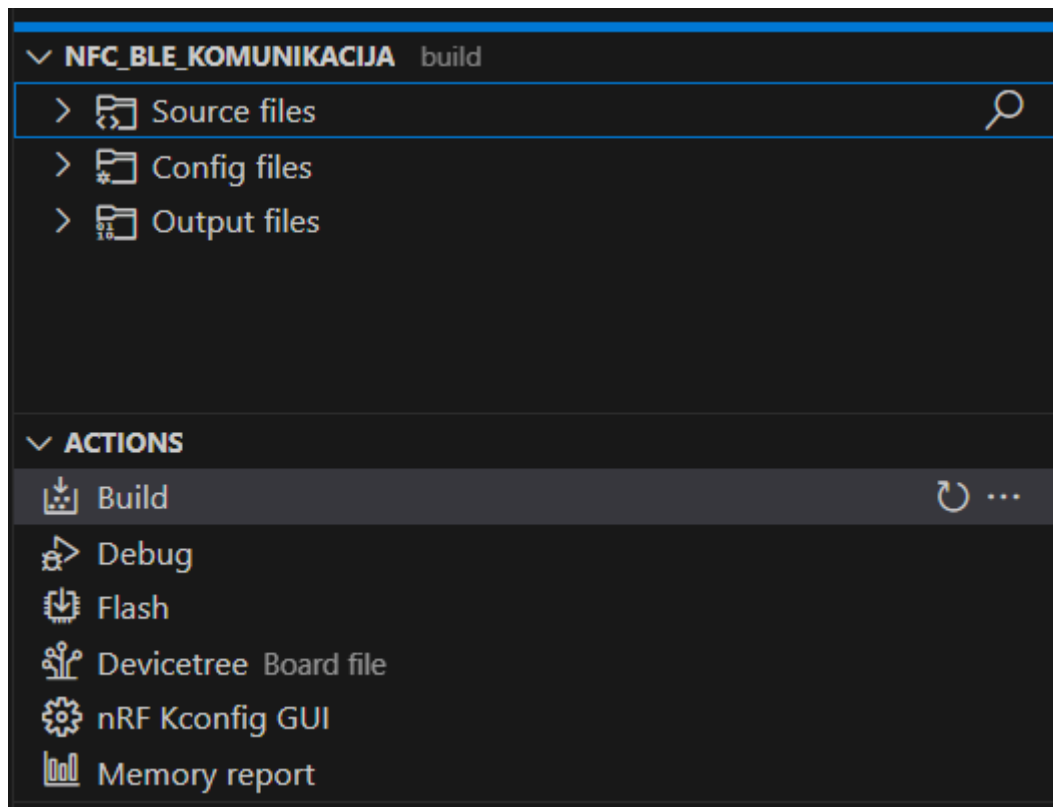
Slika 4.7. Dio prj.conf datoteke

```
1 CONFIG_REBOOT=y
2 CONFIG_FLASH=y
3 CONFIG_NFC_NDEF=y
4 CONFIG_NFC_NDEF_MSG=y
5 CONFIG_NFC_NDEF_RECORD=y
6 CONFIG_NFC_NDEF_TEXT_RECORD=y
7 CONFIG_BT_DEVICE_NAME="Nordic_nRF52840"
```



Slika 4.6. nRF Connect - build

Nakon što su svi gornji koraci upješno odrađeni vrijeme je za glavnu funkcionalnost alata nRF Connect. Pomoću ovog alata moguće je odraditi akcije build, debug i flash pritiskom na gumb pod izbornikom Actions (slika 4.8). Pritiskom na gumb build kod se priprema za izvođenje na uređaju. Pritiskom na gumb flash ponovno će se odraditi build korak te će se sav izvršni kod prebaciti na uređaj i odmah će početi sa izvođenjem. Ukoliko se odabere gumb Debug ponovno će se odraditi koraci build i flash, ali će se također pokrenuti i posebno sučelje preko kojeg je moguće izvršavati liniju po liniju napisanog koda što znatno olakšava pronalazak grešaka.



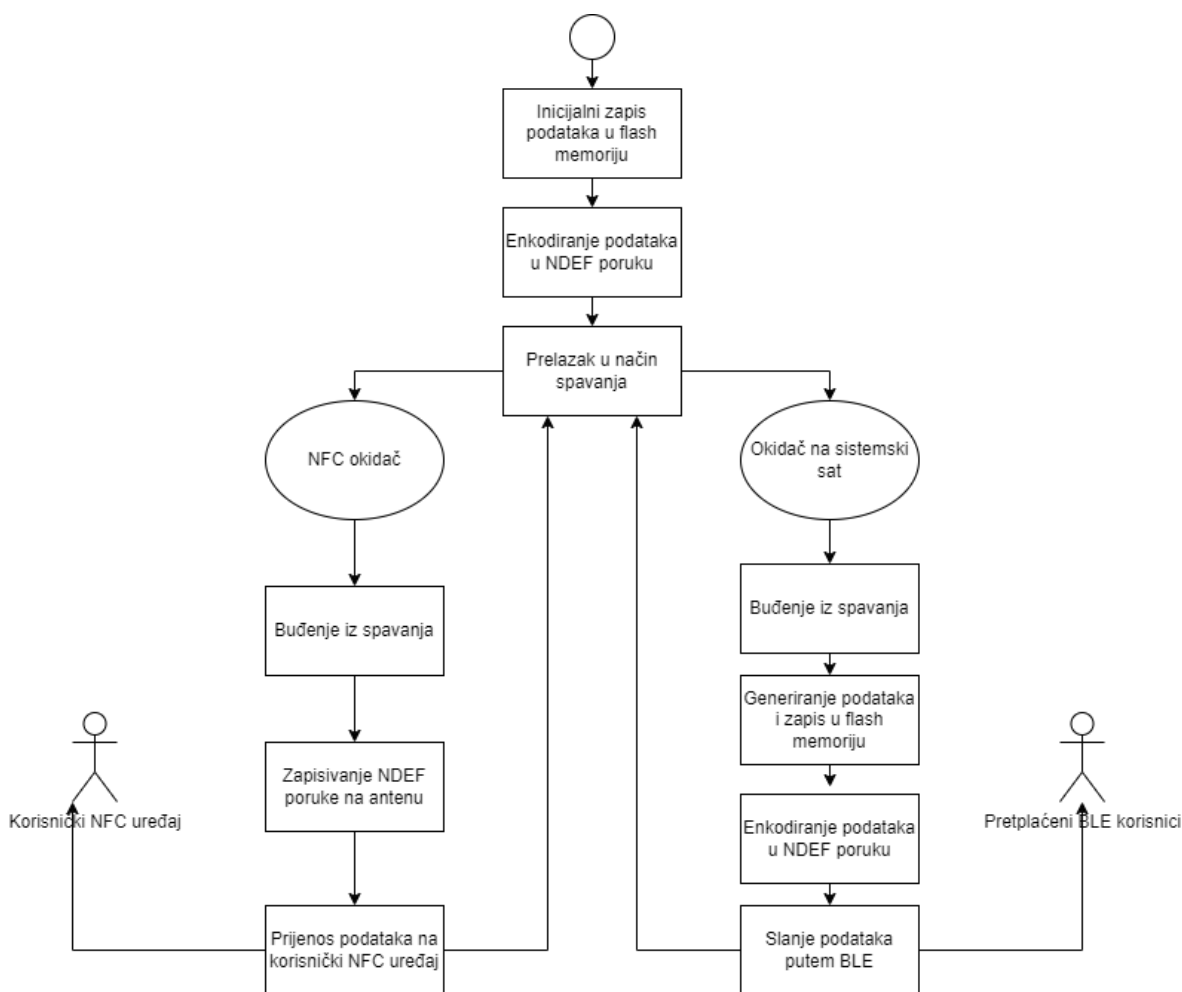
Slika 4.7. nRF Connect - flash

Programski kod

Slika 4.8 prikazuje dijagram rada programskog koda. Dijagram je podijeljen u dva glavna dijela. Prvi dio se odnosi na NFC aktivaciju. Proces započinje inicijalnim zapisom podataka u flash memoriju. Nakon toga slijedi enkodiranje tih podataka u NDEF poruku i prelazak uređaja u način spavanja kako bi se uštedjela energija. Kada NFC okidač detektira prisutnost korisničkog NFC uređaja, budi uređaj iz spavanja. Uređaj zatim zapisuje NDEF poruku na antenu i prenosi podatke na korisnički NFC uređaj. Nakon završetka NFC slanja uređaj prelazi nazad u način spavanja.

Drugi dio se odnosi na aktivaciju putem sistemskog sata. Proces također započinje inicijalnim zapisom podataka u flash memoriju i enkodiranjem tih podataka u NDEF poruku, nakon čega slijedi prelazak u način spavanja. Kada okidač na sistemskom satu aktivira uređaj, on se budi iz spavanja, generira nove podatke i zapisuje ih u flash memoriju. Potom enkodira podatke u NDEF poruku i šalje ih putem BLE (Bluetooth Low Energy) tehnologije pretplatnicima BLE. Pošto nisu korištene metode enkripcije pri uparivanju uređaja, korištene su metode uparivanja Just Works i Out of Band. Kao i kod NFC

slanja pri završetku slanja uređaj prelazi u način spavanja.



Slika 4.8. Dijagram rada programskog koda

Na početku svakog programskog koda pa tako i ovog potrebno je uključiti sve korištene knjižnice (eng. "library"). Korištene knjižnice bile su potrebne za omogućavanje ponovnog pokretanja sustava ukoliko dođe do neke neoporavljive pogreške, kao što su pogreške u inicijalnim postavljanjima parametara (primjer koda broj 1, knjižnica pod brojem 1). Zatim je bilo potrebno uključiti i knjižnice za rad sa NFC Data Exchange Format (NDEF) porukama, te općenito NFC komunikacijom (knjižnice 2-6). Kako bismo mogli ostvariti traženu funkcionalnost slanja podataka putem BLE tehnologije bilo je potrebno uključiti knjižnicu koja je dio Zephyr RTOS Bluetooth stoga i koja se koristi za upravljanje otkrivanjem GATT (Generic Attribute Profile) usluga i karakteristika (definira način na koji dva BLE uređaja razmjenjuju podatke, knjižnica broj 7). Na poslijetku kako bismo mogli koristiti LED svijetla na uređaju kao signale o uspješnosti komunikacija bilo je potrebno uključiti i knjižnicu za rad sa LED svijetlima (knjižnica broj 8).

Kod 1. Uključivanje knjižnica

```
1 #include <zephyr/sys/reboot.h>
2 #include <nfc/ndef/msg.h>
3 #include <nfc/t4t/ndef_file.h>
4 #include "ndef_file_m.h"
5 #include <nfc_t2t_lib.h>
6 #include <nfc/ndef/text_rec.h>
7 #include <bluetooth/gatt_dm.h>
8 #include <dk_buttons_and_leds.h>
```

Kako bi bilo lakše upravljati postavkama prije početka pisanja logike u kodu bitno je postaviti neke parametre. Parametri koji su definirani su interval slanja podataka putem BLE tehnologije (Kod 2, linija 1), maksimalan broj različitih poruka koje se mogu staviti na NFC oznaku (Kod 2, linija 2) koji je postavljen na 1 zato što se u jednoj poruci šalje cjelokupni set podataka. Nema potrebe razlaganja seta na više poruka jer je količina podataka koje je moguće odjednom poslati direktno vezana za veličinu buffera i stoga nema potrebe za slanjem podataka u više poruka koje su odjednom zapisane na NFC oznaci. Veličina buffera je postavljena na 1024 bytea upravo zato što je to i ograničenje koje donosi korištena NFC antena (Kod 2, linija 3). Potrebno je i definirati LED svijetla kako bi se mogli koristiti prilikom prikaza stanja sustava (Kod 2, linije 4 i 5). Pošto za potrebe ovog rada nije korišten bio nikakav senzor nego su se podaci koji su slani preko bežičnih mreža generirali u kodu bilo je potrebno definirati veličinu koja određuje koliko podataka će se u jednom životnom ciklusu rada ovog uređaja zapisati i za to je odabrana mjerna jedinica sekunde (Kod 2, linija 6).

Kod 2. Definiranje konstanti

```
1 #define SLEEP_TIME 15
2 #define MAX_REC_COUNT 1
3 #define NDEF_MSG_BUF_SIZE 1024
4 #define NFC_FIELD_LED DK_LED1
5 #define NFC_FIELD_LED_TWO DK_LED2
6 #define MAX_SECONDS_ELAPSED 1000
```

Glavna funkcija

Na prikazima koda 3-8 nalazi se glavna funkcija (main) u kojoj se pokreću sve ostale funkcije i u kojoj se nalazi glavna funkcionalnost programske potpore. Na slici Kod 3. prikazana je struktura glavne funkcije gdje se vidi kako se na samom početku konfiguriraju LED svijetla te ukoliko se ne uspiju konfigurirati program koristi "zloglasnu" goto naredbu (eng. "statement") kojom odlazi do dijela koda na kraju slike Kod 3 i gdje, ukoliko je tako omogućeno u konfiguracijskoj datoteci, ponovno pokreće ili ukoliko to nije postavljeno gasi cijeli program.

Kod 3. Struktura glavne funkcije

```
1 void main(void)
2 {
3     uint32_t len = sizeof(ndef_msg_buf);
4
5     /* Configure LED-pins as outputs */
6     if (dk_leds_init() < 0) {
7         printk("Cannot init LEDs!\n");
8         goto fail;
9     }
10
11     /*
12     Ostatak koda glavne funkcije
13     */
14
15     fail:
16     #if CONFIG_REBOOT
17         sys_reboot(SYS_REBOOT_COLD);
18     #endif
19     return;
20 }
```

Slika Kod 4. predstavlja konfiguraciju NFC Data Exchange Format zapisa kako bi mogli biti zapisani u flash memoriji uređaja te inicijalni dohvat podataka koji su već spremeni u flash memoriji kako bi se ubuduće mogli novi zapisi spremati u memoriju. Funk-

cije koje su korištene u kodu glavnog programa i koje služe kao pomoć radu sa NDEF podacima zapisane su u pomoćnoj datoteci "ndef_file_m.c". Potpuni kod tih funkcija dostupan je u dodatku X, a te funkcije su: funkcija za inicijalizaciju "ndef_file_setup", funkcija za ažuriranje NDEF podataka u flash memoriji "ndef_file_update", funkcija za postavljanje inicijalne poruke "ndef_file_default_message" te funkcija za učitavanje NDEF podataka iz flash memorije "ndef_file_load".

Kod 4. Konfiguracija NDEF-a

```
1  /* setup NDEF */
2      if (ndef_file_setup() < 0) {
3          printk("Cannot setup NDEF file!\n");
4          goto fail;
5      }
6
7      /* Load NDEF message from the flash file. */
8      if (ndef_file_load(ndef_msg_bufTwo, sizeof(ndef_msg_bufTwo)) <
9          0) {
10         printk("Cannot load NDEF file!\n");
11         goto fail;
12     }
```

Kako bi bilo moguće ostvariti slanje podataka putem protokola NFC potrebno je prvo postaviti NFC komunikaciju i callback funkciju koja će biti pozvana nakon što se okine NFC događaj (eng. "event"), odnosno nakon što se NFC oznaci približi drugi NFC uređaj, kao što je na primjer pametni telefon. Prije negoli se poruka može zapisati na NFC oznaku potrebno ju je enkodirati što je urađeno pozivom funkcije "encode_msg". Poziv funkcija za konfiguriranje callback funkcije i enkodiranje NFC poruke nalazi se na slici Kod 5., a same funkcije će biti i detaljnije objašnjene kasnije.

Kod 5. Postavljanje NFC-a

```
1  /* Set up NFC */
2      if (nfc_t2t_setup(nfc_callback_with_wake, NULL) < 0) {
3          printk("Cannot setup NFC T2T library!\n");
4          goto fail;
5      }
```



```

6
7  /* Encode message */
8  if (encode_msg(ndef_msg_buf, &len) < 0) {
9      printk("Cannot encode message!\n");
10     goto fail;
11 }

```

Nakon što su željene poruke enkodirane i nakon što je NFC konfiguracija odrađena potrebno je postaviti poruku, odnosno "payload" na NFC oznaku te pokrenuti emulaciju NFC-a funkcijom prigodnog naziva "nfc_t2t_emulation_start" nakon čijeg poziva je program spreman za slanje poruka putem NFC protokola.

Kod 6. Postavljanje NFC poruke i početak oglašavanja

```

1  /* Set created message as the NFC payload */
2  if (nfc_t2t_payload_set(ndef_msg_buf, len) < 0) {
3      printk("Cannot set payload!\n");
4      goto fail;
5  }
6
7  /* Start sensing NFC field */
8  if (nfc_t2t_emulation_start() < 0) {
9      printk("Cannot start emulation!\n");
10     goto fail;
11 }

```

Nakon postavljanja NFC protokola, preostalo je još postaviti i BLE te pokrenuti emitiranje BLE signala za povezivanje s drugim uređajima. Za navedene akcije postoje već implementirane metode koje je samo potrebno pozvati sa odgovarajućim parametrima. Kako bi dobili funkcionalnost periodičnog slanja podataka potrebno je i postaviti brojač vremena koji koristi ranije spomenuti parametar "SLEEP_TIME" koji određuje u kojim intervalima će se željene poruke upravo i slati svim uređajima koji su sa sustavom spojeni putem protokola BLE. Navedene implementacije nalaze se na slici Kod 7.

Kod 7. BLE

```

1  /* Initialize the BLE Subsystem */
2      int errBle = bt_enable(NULL);
3      if (errBle) {
4          goto fail;
5      }
6
7      /* Start advertising */
8      errBle = bt_le_adv_start(BT_LE_ADV_CONN_NAME, ad, ARRAY_SIZE(ad
9      ), NULL, 0);
10     if (errBle) {
11         goto fail;
12     }
13
14     /* Start the timer for periodic BLE data sending */
15     k_timer_start(&ble_send_timer, K_SECONDS(SLEEP_TIME), K_SECONDS
(SLEEP_TIME));

```

Nakon uspješno odrađenih postavljanja potrebnih parametara, konfiguriranja NDEF formata zapisa te inicijalizacija BLE i NFC protokola potrebno je pokrenuti beskonačnu petlju u kojoj će se odvijati sva buduća zapisavanja podataka na NFC oznaku te slanja podataka putem NFC i BLE protokola. Kako bi bilo moguće pokrenuti proces enkodiranja te zapisivanja NFC poruka prvo je iznimno bitno zaustaviti emulaciju jer u suprotnom ne će biti moguće zapisati podatke na NFC oznaku (Kod 8, linija 3). Nakon što je ugašena emulacija istim procesom kao i prije beskonačne petlje radi se enkodiranje i zapisivanje poruke na NFC oznaku te spremanje podataka podataka u flash memoriju. Radi lakšeg procesa pronalazaka grešaka dodano je i čitanje zapisanih podataka (Kod 8, linija 13). Nakon što su željeni podaci zapisani potrebno je ponovno upaliti NFC emulaciju, a radi što veće uštede energije uređaja pokreće se i način rada u kojemu uređaj beskonačno miruje. Postavljeno je da uređaj beskonačno miruje umjesto da miruje određeni interval iz razloga što je na brojaču vremena već definirano nakon koliko vremena će probuditi uređaj iz stanja mirovanja te nema potrebe za redefiniranjem.

Kod 8. Beskonačna petlja unutar glavne funkcije

```

1  while (1) {

```

```

2      // System goes to sleep until next event (timer or NFC)
3      nfc_t2t_emulation_stop();
4      static uint32_t len = NDEF_MSG_BUF_SIZE;
5      if (encode_msg(ndef_msg_buf, &len) < 0) {
6          printk("Cannot encode message!\n");
7      } else {
8          int ret = nfc_t2t_payload_set(ndef_msg_buf, len);
9          if (ret < 0) {
10             printk("Cannot set payload!\n");
11         }
12     }
13     if (ndef_file_load(ndef_msg_bufTwo, sizeof(ndef_msg_bufTwo)
14 ) < 0) {
15         printk("Cannot load NDEF file!\n");
16     }
17     if (ndef_file_update(ndef_msg_bufTwo, sizeof(
18 ndef_msg_bufTwo)) < 0 ) {
19         printk("Cannot update\n");
20     }
21     nfc_t2t_emulation_start();
22     k_sleep(K_FOREVER);
23 }

```

Pomoćne funkcije

U ovom radu odabrano je da se kao poruke šalju nizovi znakova (string) kao generirani podaci. Ukoliko bi htjeli slati podatke učitane sa stvarnog senzora mogli bi se slati na identičan način tako da se učitani podaci, koji su na primjer cjelobrojnog tipa (integer), prvo pretvore u znakovni niz te se zatim šalju na identičan način kao što je pokazano. Format podataka koji je odabran podrazumijeva da svaka poruka koja se šalje počinje izrazom "Payload:" i stoga je prvi korak u enkodiranju poruka dodavanje stringa "Payload:" na početak poruke. Inicijalni string "Payload:" zapisan je u varijabli "initialPayload" (Kod 9, linija 3). Generirana poruka koristi UTF-8 kodiranje za tekst i jezični kod "en_code". Veličina jezičnog koda se određuje pomoću izraza "sizeof(en_code)" dok je veličina ukup-

nog zapisa dinamična, što znači da može varirati u duljini, ovisno o veličini poruke koju se želi poslati.

Kod 9. Enkodiranje NFC poruke

```
1 static int encode_msg(uint8_t *buffer, uint32_t *len)
2 {
3     snprintf(dynamic_payload, sizeof(dynamic_payload), "%s",
4             initialPayload);
5
6     NFC_NDEF_TEXT_RECORD_DESC_DEF(nfc_en_text_rec, UTF_8, en_code,
7     sizeof(en_code), dynamic_payload, strlen(dynamic_payload));
8     NFC_NDEF_MSG_DEF(nfc_text_msg, MAX_REC_COUNT);
9
10    int err = nfc_ndef_msg_record_add(&NFC_NDEF_MSG(nfc_text_msg),
11    &NFC_NDEF_TEXT_RECORD_DESC(nfc_en_text_rec));
12
13    if (err < 0) {
14        return err;
15    }
16
17    err = nfc_ndef_msg_encode(&NFC_NDEF_MSG(nfc_text_msg), buffer,
18    len);
19    return err;
20 }
```

Kao što je ranije spomenuto kako bi NFC komunikacija bila ostvarena potrebno je definirati callback funkciju (Kod 10). Callback funkcija "nfc_callback_with_wake" se sastoji od poziva na drugu callback funkciju "nfc_callback" koja odrađuje sav bitan posao vezan uz NFC komunikaciju i dodatnom logikom za buđenje sustava iz stanja mirovanja. Funkcija "nfc_callback_with_wake" pobuđuje sustav zaustavljanjem brojača vremena (timera) te ponovno vraća sustav u stanje mirovanja ponovnim pokretanjem brojača.

Kod 10. NFC Callback Wrapper

```
1 void nfc_callback_with_wake(void *context, nfc_t2t_event_t event,
2     const uint8_t *data, size_t data_length)
```

```

2 {
3     nfc_callback(context, event, data, data_length);
4
5     k_timer_stop(&ble_send_timer); // Stop the timer to wake up
the system
6     k_timer_start(&ble_send_timer, K_SECONDS(SLEEP_TIME), K_SECONDS
(SLEEP_TIME)); // Restart the timer
7 }

```

Funkcija "nfc_callback" aktivira se prilikom svakog okinutog NFC događaja. Kada se vanjski NFC uređaj (npr. pametni telefon) približi NFC oznaci i aktivira NFC također se aktivira i LED lampica broj 1 (Kod 11, linija 9), a nakon što se vanjski NFC uređaj ukloni ta ista LED lampica se ugasi (Kod 11, linija 13). Korisnicima je prisutnost goreće LED lampice rednog broja 1 dodatan indikator uspješnosti slanja poruke (ona se pali isključivo u slučaju detektiranja prisutnosti drugog NFC uređaja), a ukoliko je došlo do pogreške kao indikator svijetli LED lampica broj 2.

Kod 11. NFC Callback

```

1 static void nfc_callback(void *context, nfc_t2t_event_t event,
const uint8_t *data, size_t data_length)
2 {
3     ARG_UNUSED(context);
4     ARG_UNUSED(data);
5     ARG_UNUSED(data_length);
6
7     switch (event) {
8     case NFC_T2T_EVENT_FIELD_ON:
9         dk_set_led_on(NFC_FIELD_LED);
10        break;
11
12    case NFC_T2T_EVENT_FIELD_OFF:
13        dk_set_led_off(NFC_FIELD_LED);
14        break;
15

```

```

16     default :
17         break ;
18     }
19 }

```

Kako bi se na svaki završetak odbrojanja brojača vremena poslala poruka putem BLE tehnologija potrebno je prijaviti funkciju koja će izvršiti slanje. Prijavlivanje funkcije prikazano je u Kodu 12, a prijavljena funkcija zove se "ble_send_work_handler".

Kod 12. Prijava funkcije za slanje putem BLE

```

1 K_WORK_DEFINE( ble_send_work , ble_send_work_handler );

```

Funkcija koja obrađuje logiku slanja poruka povodom isteka brojača vremena sastoji se od 2 dijela. Prvi dio funkcije generira podatke koji će biti poslani na temelju vremena (za svaku sekundu koju sustav provede izvan načina male potrošnje energije dodaje broj jednak proteklim sekundama u polje. Zatim u varijablu temp zapisuje podatke o svim proteklim sekundama odvojene znakom ";" te ih spaja sa nizom znakova "Payload:" koji služi kao prefiks (Kod 13, linije 12 i 13).

Drugi dio koda ove funkcije gasi emulaciju NFC-a te enkodira i postavlja podatke na NFC oznaku (Kod 13, linije 14 - 24). Nakon što pripremi podatke za slanje putem NFC-a sprema pripremljene podatke u flash memoriju uređaja i ponovno pokreće emulaciju NFC-a.

Kod 13. Funkcija za slanje putem BLE

```

1 static void ble_send_work_handler( struct k_work *work)
2 {
3     static uint32_t len = sizeof(ndef_msg_buf);
4     char temp[20];
5
6     if (seconds_elapsed < MAX_SECONDS_ELAPSED) {
7         seconds_elapsed++;
8         seconds_list[seconds_count++] = seconds_elapsed;
9     }
10

```

```

11  snprintf(temp, sizeof(temp), "%d;", seconds_elapsed);
12  strcat(initialPayload, temp);
13  len = NDEF_MSG_BUF_SIZE;
14  nfc_t2t_emulation_stop();
15  if (encode_msg(ndef_msg_buf, &len) < 0) {
16      dk_set_led_on(NFC_FIELD_LED_ERROR);
17      printk("Cannot encode message!\n");
18  } else {
19      int ret = nfc_t2t_payload_set(ndef_msg_buf, len);
20      if (ret < 0) {
21          dk_set_led_on(NFC_FIELD_LED_ERROR);
22          printk("Cannot set payload!\n");
23      }
24  }
25
26  // Prepare the flash buffer with the current message
27  flash_buffer_prepare(len);
28
29  if (ndef_file_update(flash_buf, flash_buf_len) < 0) {
30      printk("Cannot update NDEF file\n");
31      return;
32  }
33
34  if (ndef_file_load(ndef_msg_bufTwo, sizeof(ndef_msg_bufTwo)) <
35  0) {
36      printk("Cannot load NDEF file!\n");
37      return;
38  }
39  nfc_t2t_emulation_start();

```

5. Zaključak

U ovom radu uspješno je implementiran pametni senzor male potrošnje temeljen na nRF52840 sustavu na čipu tvrtke Nordic Semiconductor. Senzor omogućuje mjerenje, pohranu te prijenos podataka uz visoku energetska učinkovitost korištenjem BLE i NFC tehnologija. Implementacija senzora uključuje konfiguraciju sklopovskih i programskih komponenti, uključujući enkodiranje NDEF poruka i postavljanje BLE oglašavanja. Pomoću nRF Connect alata unutar programa za uređivanje programskog koda "Visual Studio Code" prenešena je razvijena programska potpora na sklopovlje. Implementirani sustav moguće je unaprijediti korištenjem sigurnosnih komponenti, kao što su kriptiranje podataka jer se trenutno podaci ne kriptiraju i javno su dostupni. Kao usko grlo prilikom NFC komunikacije pokazala se korištena NFC antena koja je simulirala NFC oznaku koja je kapaciteta od samo 1024 bytea te bi se cijeli sustav znatno unaprijedio ukoliko bi se koristio pristup koji bi omogućio slanje većih datoteka ili ukoliko bi se integriralo slanje uzajamnih manjih poruka i na taj način prenio veći skup podataka.

Literatura

- [1] ARM, <https://www.arm.com/products/silicon-ip-cpu/cortex-m/cortex-m7>, [mrežno; stranica posjećena: lipanj 2024.].
- [2] Nordic Semiconductor, https://infocenter.nordicsemi.com/pdf/nRF52840_OPS_v0.5.pdf, [mrežno; stranica posjećena: lipanj 2024.].
- [3] —, https://infocenter.nordicsemi.com/topic/struct_sdk/struct/sdk.html, [mrežno; stranica posjećena: lipanj 2024.].
- [4] https://technotes.kynetics.com/2018/BLE_Pairing_and_bonding/, [mrežno; stranica posjećena: lipanj 2024.].
- [5] https://play.google.com/store/apps/details?id=com.wakdev.wdnfc&hl=en_US&pli=1, [mrežno; stranica posjećena: lipanj 2024.].

Sažetak

Pametni senzor male potrošnje

Matija Turkalj

U današnjem svijetu potreba za senzorima male potrošnje energije postaje sve važnija. S obzirom na rastući trend povezivanja sve više različitih uređaja putem interneta u internetu stvari (IoT), smanjena potrošnja energije postala je ključna karakteristika ovih uređaja. Ovaj diplomski rad fokusira se na razvoj pametnog senzora male potrošnje korištenjem razvojnog sustava zasnovanog na čipu nRF52840. Glavni cilj ovog rada je demonstrirati kako kreirati senzor koji ne samo da učinkovito mjeri i prenosi podatke putem Bluetooth Low Energy (BLE) i Near Field Communication (NFC) tehnologija, već i optimizira svoju potrošnju energije kroz različite radne modove.

Ključne riječi: Near Field Communication (NFC); Energetska učinkovitost; System on Chip (SoC); Bluetooth Low Energy (BLE); nRF52840; ARM Cortex-M4 procesor

Abstract

Pametni senzor male potrošnje

Matija Turkalj

In today's world the need for low-power consumption sensors is becoming increasingly important. Given the growing trend of connecting more and more diverse devices through the Internet in the Internet of Things (IoT), reduced energy consumption has become a key feature of these devices. This thesis focuses on the development of a smart low-power sensor using a development system based on the nRF52840 chip. The main goal of this work is to demonstrate how to create a sensor that not only efficiently measures and transmits data via Bluetooth Low Energy (BLE) and Near Field Communication (NFC) technologies, but also optimizes its energy consumption through various operating modes.

Keywords: Near Field Communication (NFC); Energy efficiency; System on Chip (SoC); Bluetooth Low Energy (BLE); nRF52840; ARM Cortex-M4 processor